

## **AMENDMENTS TO THE CLAIMS**

This listing of claims replaces all prior versions, and listings, of claims in the application:

### **Listing of Claims:**

1. (Original): A computer-implemented method for an object-oriented environment comprising:

wrapping a reference to a second object within a second context with a proxy wrapper, the second context defining at least a second set of arbitrary invariants on a second set of arbitrary objects including the second object;

calling of the second object within the second context by a first object within a first context via the reference as wrapped in the proxy wrapper, the first context defining at least a first set of arbitrary invariants on a first set of arbitrary objects including the first object; and

returning of the second object within the second context to the first object within the first context via the reference as wrapped in the proxy wrapper.

2. (Original): The method of claim 1, further comprising:

wrapping a reference to the first object within the first context with a proxy wrapper;

calling of the first object by the second object via the reference as wrapped in the proxy wrapper; and,

returning of the first object to the second object via the reference as wrapped in the proxy wrapper.

3. (Original): The method of claim 1, further comprising:

calling of a third object that is agile by the first object via an unwrapped direct reference, such that the first context of the first object becomes a context for the third object for calling of the third object by the first object, such that the third object executes in the first context of the first object, and such that the agile object is agile in that the agile object has no permanent context; and,

responding by the third object to the first object via the unwrapped direct reference.

4. (Original): The method of claim 3, further comprising:  
calling of the first object by the third object via an unwrapped direct reference; and,  
responding by the first object to the third object via the unwrapped direct reference.

5. (Original): The method of claim 1, further comprising:  
calling of a third object within the first context by the first object via an unwrapped dir  
reference; and,  
responding by the third object to the first object via the unwrapped direct reference.

6. (Currently Amended): A computerized system comprising:

a processor; and

computer-readable memory storing:

at least one first object within a first context, the first context defining at least a first set of arbitrary invariants on a first set of arbitrary objects including the at least one first object; and

at least one second object within a second context, the second context defining at least a second set of arbitrary invariants on a second set of arbitrary objects including the at least one second object,

such that the at least one first object communicates with one another directly via direct references, the at least one second object communicates with one another directly via direct references, and any of the at least one first object communicates with any of the at least one second object via indirect references wrapped in proxy wrappers.

7. (Previously Presented): The system of claim 6, further comprising at least one agile object, such that the at least one agile object is agile in that the at least one agile object has no permanent context.

8. (Previously Presented): The system of claim 7, wherein as called by any of the at least one first object any of the at least one agile object executes within the first context, such that any of the at least first object communicates with any of the at least one agile object directly.

9. (Previously Presented): The system of claim 7, wherein as called by any of the at least one second object any of the at least one agile object executes within the second context, such that any of the at least second object communicates with any of the at least one agile object directly.

10. (Original): The system of claim 6, wherein a reference to one of the at least one second object is wrapped in a proxy wrapper, and one of the at least one first object calls the one of the at least one second object via the reference as wrapped in the proxy wrapper.

11. (Original): The system of claim 6, wherein a reference to one of the at least one first object is wrapped in a proxy wrapper, and one of the at least one second object calls the one of the at least one first object via the reference as wrapped in the proxy wrapper.

12. (Currently Amended): A computerized system comprising:

a processor; and

computer-readable memory storing:

at least one first object within a first context, the first context defining at least a first set of arbitrary invariants on a first set of arbitrary objects including the at least one first object, the at least one first object communicating with one another directly via direct references;

at least one second object within a second context, defining at least a second set of arbitrary invariants on a second set of arbitrary objects including the at least one second object, the at least one second object communicating with one another directly via direct references, and any of the at least one first object communicating with any of the at least one second object via indirect references wrapped in proxy wrappers; and[[i]]

at least one agile object, such that the at least one agile object are agile in that the at least one agile object have no permanent context, such as called by any of the at least one first object any of the at least one agile object executes within the first context, and such that as called by any of the at least one second object any of the at least one agile object executes within the second context.

13 (Currently Amended): A machine-readable storage medium having a computer program stored thereon for execution by a processor to perform a method comprising:

wrapping a reference to a second object with a second context with a proxy wrapper, the second context defining at least a second set of arbitrary invariants on a second set of arbitrary invariants on a second set of arbitrary objects including the second object;

calling of the second object within the second context by a first object within a first context via the reference as wrapped in the proxy wrapper, the first context defining at least a first set of arbitrary invariants on a first set of arbitrary objects including the first object; and[[;]]

returning of the second object within the second context to the first object within the first context via the reference as wrapped in the proxy wrapper.

14. (Original): The medium of claim 13, further comprising:

wrapping a reference to the first object within the first context with a proxy wrapper;

calling of the first object by the second object via the reference as wrapped in the proxy wrapper; and,

returning of the first object to the second object via the reference as wrapped in the proxy wrapper.

15. (Original): The medium of claim 13, further comprising:

calling of a third object that is agile by the first object via an unwrapped direct reference, such that the first context of the first object becomes a context for the third object for calling of the third object by the first object, such that the third object executes in the first context of the first object, and such that the agile object is agile in that the agile object has no permanent context; and

responding by the third object to the first object via the unwrapped direct reference.

16. (Original): The medium of claim 15, further comprising: calling of the first object by the third object via an unwrapped direct reference; and, responding by the first object to the third object via the unwrapped direct reference.

17. (Original): The medium of claim 13, further comprising:

calling of a third object within the first context by the first object via an unwrapped direct reference; and,

responding by the object to the first object via the unwrapped direct reference.